A 1.5-μW Fully-Integrated Keyword Spotting SoC in 28-nm CMOS With Skip-RNN and Fast-Settling Analog Frontend for Adaptive Frame Skipping

Heejin Yang, *Graduate Student Member, IEEE*, Ji-Hwan Seol, *Member, IEEE*, Rohit Rothe, *Graduate Student Member, IEEE*, Zichen Fan, *Graduate Student Member, IEEE*, Qirui Zhang, *Member, IEEE*, Hun-Seok Kim, *Senior Member, IEEE*, David Blaauw, *Fellow, IEEE*, and Dennis Sylvester, *Fellow, IEEE*

***Abstract*— We propose a fully integrated low-power keyword spotting (KWS) system on chip (SoC) with content-adaptive frame subsampling, implemented in 28-nm CMOS technology. The system is co-optimized from end-to-end including the analog frontend (AFE) and digital backend with a skip-recurrent neural network (RNN) KWS algorithm. The SoC performs dynamic power gating based on the decision from the skip-RNN algorithm that allows opportunistic frame skipping to reduce the power consumption without compromising the KWS accuracy. The design employs a fast-stabilizing AFE, enabling fast OFF to ON transitions with a settling time of less than 1 ms. A low-power feature extractor (FE) and RNN classifier sprint with a relatively fast clock to minimize the latency of the frame-skipping decision and to minimize the leakage power overhead. The SoC integrates a custom-designed latch-based always-on ON-chip memory to reduce leakage power to store all RNN weights on the chip. The proposed system achieves 1.48 μW with an average of 76% skip ratio across frames, achieving 92.8% accuracy on a 7-class subset of the GSCD dataset. This work represents a significant step toward a low-power KWS SoC with content-adaptive frame subsampling for energy-efficient, deep-learning-enabled Internet-of-Things (IoT) devices.**

***Index Terms*— Fast settling analog frontend (AFE), keyword spotting (KWS), mel frequency cepstral coefficient (MFCC), neural network (NN) accelerator, skip-recurrent NN (RNN).**

I. Introduction

SPEECH is a natural way of communication for humans and, thus, has become a prevalent and important modality to interface between humans and machines in the era of the Internet of Things (IoT). With the recent advances in deep learning, the accuracy of speech recognition has been greatly improved. As a result, a fast-growing number of devices are adopting deep-learning-based speech recognition as their interface [2]. Traditionally, due to the large processing workload and complexity associated with deep-learning algorithms, speech recognition tasks have been primarily executed in the cloud [3]. However, this approach raises several issues [4]. First, with the increased number of connected devices to the cloud, bandwidth resources can be highly congested because of the large amount of data generated from the connected edge/IoT devices. Second, there are security concerns related to always uploading private data to the cloud.

To address the aforementioned challenges, significant efforts have been made recently to move data processing from the cloud to the edge. One such application is the keyword spotting (KWS) wake-up system on the edge. The KWS system continuously monitors the audio input for specific keywords and subsequently activates the rest of the system or enables more complicated speech recognition tasks when detecting the keywords. Commercialized examples of the KWS systems include Amazon Alexa and Apple Siri.

Since the KWS system should be running continuously, its power consumption is of primary concern to maximize the battery life of the system. At the same time, the accuracy of the KWS system is also important for a better user experience. However, designing the KWS system with lower power is not an easy task due to the complexity of codesigning the analog frontend (AFE) and digital backend to execute a sophisticated neural network (NN) algorithm.

Recently, a large amount of work dedicated to KWS has been proposed, especially targeting low power consumption for running on energy-constrained IoT edge devices. A prior design [5] can execute real-time KWS with 300-μW power by using compact memory storage for all ON-chip weights of 270 KB and a programmable processor. An single instruction multiple data (SIMD) processor proposed by [6] utilizes a deep neural network (DNN) to run a small vocabulary recognition task consuming 172 μW. The design in [7] employs a binary NN and depth-wise convolution to achieve a low power consumption of 510 nW.

However, these previously proposed designs have primarily focused on reducing the power consumption of the digital backend while neglecting the AFE, which often dominates

Manuscript received 5 May 2023; revised 9 July 2023 and 24 August 2023; accepted 7 September 2023. Date of publication 3 October 2023; date of current version 29 December 2023. We confirm that our sponsors are as stated in the acknowledgment. (TSMC University Shuttle Program, SRC Cognisense JUMP 2.0). This article was approved by Associate Editor Visvesh Sathe. This work was in part published at ISSCC 2023 [1]. *(Heejin Yang and Ji-Hwan Seol are co-first authors.) (Corresponding author: Heejin Yang.)*

Heejin Yang, Rohit Rothe, Zichen Fan, Qirui Zhang, Hun-Seok Kim, David Blaauw, and Dennis Sylvester are with the Department of Electrical and Computer Engineering, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: heejiny@umich.edu).

Ji-Hwan Seol is with the Department of Electrical and Computer Engineering, University of Michigan, Ann Arbor, MI 48109 USA, and also with Samsung Electronics, Hwaseong, Gyeonggi 18448, South Korea.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JSSC.2023.3316648.

Digital Object Identifier 10.1109/JSSC.2023.3316648

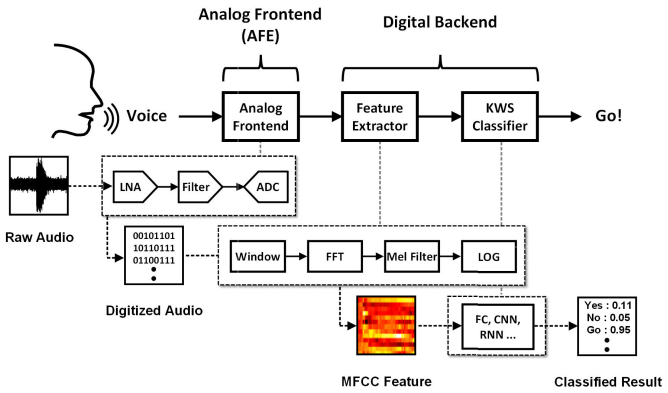


Fig. 1. Basic KWS system pipeline.

the overall power consumption in the actual KWS system. The AFE includes power-depending components such as a low-noise audio amplifier and analog-to-digital converter (ADC) to receive an incoming raw audio signal from a microphone and convert it to digital form so that the digital backend can process it. Other prior works that tried optimization of the end-to-end power consumption [8], [9], consume more than 10 μWs, which remains too high to assist small IoT systems with constrained battery capacity [10]. Thus, an end-to-end KWS system requires jointly optimizing the AFE and digital backend for overall power minimization.

This work proposes a fully integrated KWS system where the AFE and the digital backend are co-optimized. The AFE consists of a low-noise amplifier (LNA), an analog filter, and an ADC, whereas the digital backend integrates a feature extractor (FE) and recurrent NN (RNN) classifier. The proposed system employs an adaptive NN, so-called skip-RNN, which makes the content-adaptive decision to opportunistically skip audio frames whenever possible to minimize the power consumption [1].

Our contributions are summarized as follows.

1) Demonstration of the skip-RNN algorithm [11] to adaptively power gate (PG) both the AFE and the digital backend based on the input context, achieving 3× power reduction in total.
2) Introducing an AFE structure that features a dc-coupled first stage with a switched capacitive feedback (CF) resistor to enable fast off-to-on transitions with a settling time of less than 1 ms.
3) Designing and evaluating an FE and NN classifier that employs computational sprinting with efficient scheduling to reduce their operation time and static current.
4) Proposing an FE that implements fast Fourier transform (FFT) with four bank single-port SRAMs and power-optimized computations and controls.
5) Designing a low-leakage custom memory for always-on weight memory (WMEM) to minimize the NN classifier power consumption.

II. KWS System Overview and Challenges

A. Basic KWS System Pipeline

Fig. 1 illustrates the pipeline of a typical KWS system. First, the voice signal from the microphone is applied to the AFE, passing a low-noise amplifier and a filter to get the desired speech signal with reduced noise. Subsequently, an ADC converts the continuous signal into quantized digital samples.

The audio samples are then fed into the FE module to produce audio feature vectors, which are used as the input to the KWS classifier. This feature extraction step reduces the input dimensionality and improves the accuracy of the classifier. Among various FE algorithms, mel frequency cepstral coefficient (MFCC) is the most common choice [12] which is also adopted in this work. The FE first segments the long audio signal into T short-time audio frames. Then, a windowing function is applied to each frame to prevent spectral leakage. After that, the FFT is performed and the power spectrum is computed for each frame. Finally, the Mel filter, which is a triangular filter that converts the frequency to the Mel frequency domain, and the log filter are applied to generate the $T \times F$ dimension feature vector.

The KWS classifier receives the feature vector from the FE and generates probabilities of each keyword in the audio signal. In recent years, NN-based KWS classifiers have become the mainstream approach. For instance, [13] uses a spiking NN (SNN)-based classifier using a spike-rate coded FE. The design in [14] adopts a small-size convolutional NN (CNN) along with an event-driven FE exhibiting ultralow power consumption. Some prior designs such as [7] and [15] use a depthwise separable CNN (DS-CNN), an efficient alternative to a conventional CNN, as a classifier. Other works [8], [9], [16], [17], [18] utilize the RNN which shows high accuracy even with a small-sized network by exploiting the temporal dependencies of audio signals.

A comparison of the performance of various NN models with different network sizes was conducted in [19]. The study in [19] shows that DS-CNN exhibits the best accuracy. However, the same study shows that the RNN families require a fewer number of operations for small-sized models compared to CNNs (including DS-CNN). Also, in terms of accuracy, RNNs typically outperform conventional CNNs and are only slightly worse (less than 1% accuracy difference with a similar memory footprint) than DS-CNN. Since our primary goal is to minimize power consumption while maintaining competitive accuracy, we select a (skip-)RNN as our algorithm target as it can provide additional power savings in the AFE by applying content-adaptive opportunistic frame skipping.

B. Challenges

Designing a low-power KWS system suitable for small IoT devices with limited battery capacity is not straightforward. For example, assuming the standard SR63 battery with 1.55 V supply and 17 mAh, KWS power consumption should be lower than approximately 1.5 μW to guarantee battery life exceeding two years. This requires optimization of both the analog and digital parts of the system to minimize overall power consumption.

As mentioned in Section I, the AFE consumes a considerable amount of power. Since the incoming microphone signal can be very small and it is crucial to minimize the additional noise and distortion to the audio signal, the power consumption of the amplifier specifically can be significant. For example,

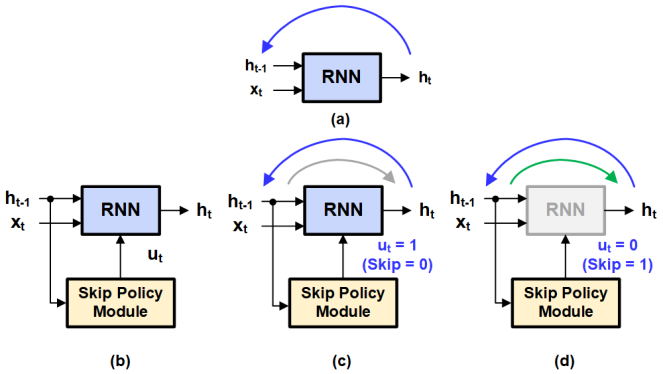


Fig. 2. Conceptual diagram of (a) conventional RNN and (b) skip-RNN. (c) Skip-RNN when the skip signal is 0. (d) Skip-RNN when the skip signal is 1.

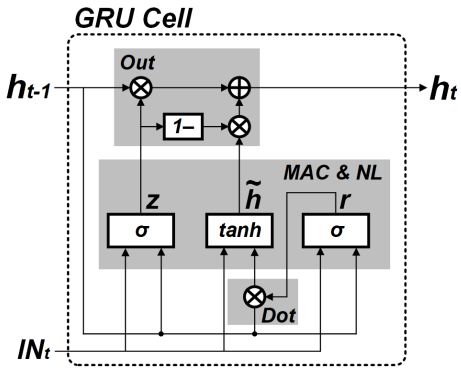


Fig. 3. GRU cell implementation.

standard OFF-chip microphone preamplifiers typically consume hundreds of μW [20], [21]. Even custom-designed integrated amplifiers that dissipate lower power often still consume several μW to tens of μW [22].

MFCC FE is another power-hungry component. While the algorithm itself can be easily implemented in hardware, the complex computations involved, especially in the FFT module, lead to high power consumption. Reducing the power and memory size of the NN classifier while maintaining high accuracy is another challenge. Even though the RNN-based classifier requires a smaller memory size than that of a CNN-based classifier with similar accuracy, a naive implementation would consume a few hundred μW. Thus, a specialized hardware accelerator is needed for the digital backend to enable KWS within our power budget ($\leq$1.5 μW).

III. Proposed KWS System Optimization: System and Algorithm Level

A. Skip-RNN

This work adopts the skip-RNN algorithm [11] to reduce system energy in both analog and digital blocks. Skip-RNN is a type of dynamic NN that can adapt its computational graph based on the input. This allows an efficient allocation of computing resources, saving computational energy. Specifically, skip-RNN generates a skip signal at each time step, alongside the classification result. This is used to determine whether or not the computation for the next time step should be skipped.

The mathematical formulation of skip-RNN is as follows, building upon the definition of the conventional RNN. Given an input feature vector x_t, the output of a conventional RNN at time step t is the hidden state vector h_t. In the following time step $t+1$, the previous output h_t is fed back. This is summarized in Fig. 2(a) and the following equation:

$$h_t = S(h_{t-1}, x_t). \tag{1}$$

Skip-RNN is a combination of a conventional RNN and the skip policy module, which is also another form of NN [see (2)]. The output of the skip policy module is a binary state update gate $u_t \in \{0, 1\}$ [Fig. 2(b)]. When $u_t = 1$ as depicted in Fig. 2(c), h_{t-1} is updated to the new output h_t, operating in the same way as the conventional RNN does. However, when the $u_t = 0$, as shown in Fig. 2(d), the main RNN skips its operation and simply feedforward its input, which is the output of the previous time step. By reducing the number of hidden stage updates in this way, skip-RNN can achieve faster inference and reduced energy consumption compared to its conventional counterpart

$$h_t = u_t \cdot S(h_{t-1}, x_t) + (1 - u_t) \cdot h_{t-1}. \tag{2}$$

In [11], at every time step, the skip policy module computes $\tilde{u}_t \in [0, 1]$, which is the probability of performing the state update at the next time step, as follows:

$$u_t = f_{\text{binarize}}(\tilde{u}_t) \tag{3}$$

$$\Delta\tilde{u}_t = \text{sigmoid}(W_u \cdot h_t + b_u) \tag{4}$$

$$\tilde{u}_{t+1} = u_t \cdot \Delta\tilde{u}_t + (1 - u_t) \cdot \left(\tilde{u}_t + \min\left(\Delta\tilde{u}_t, 1 - \tilde{u}_t\right)\right). \tag{5}$$

In our implementation, instead of using a stochastic function that requires a random number generator and associated hardware cost, we decided to derive u_t from $\tilde{u}_t$ with a simple rounding function ($u_t = \text{round}(\tilde{u}_t)$), so that the number of skip cycles is deterministic and computed once for all consecutive skips. The weights W_u and b_u can be trained using backpropagation, as opposed to other dynamic NNs where reinforcement learning is typically used with a relatively long training time.

B. RNN Cell Implementation and Hyperparameter Decision

The gated recurrent unit (GRU) is chosen as the RNN cell because it demonstrates higher accuracy and faster convergence time compared to long short-term memory (LSTM) with the same number of parameters, as experimentally shown in [23]. Fig. 3 illustrates the GRU cell, which consists of an input feature vector x_t, current hidden state vector h_{t-1}, candidate hidden state vector $\tilde{h}_t$, and two gates: the reset gate r_t and update gate z_t. The output of the GRU cell is represented as follows:

$$r_t = \text{sigmoid}\left(W_r \cdot \left[h_{t-1}, x_t\right] + b_r\right) \tag{6}$$

$$z_t = \text{sigmoid}\left(W_z \cdot \left[h_{t-1}, x_t\right] + b_z\right) \tag{7}$$

$$\tilde{h}_t = \tanh\left(W_h \cdot \left[r_t \odot h_{t-1}, x_t\right] + b_h\right) \tag{8}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t \tag{9}$$

where W and b are trainable parameters that are stored in the WMEM.

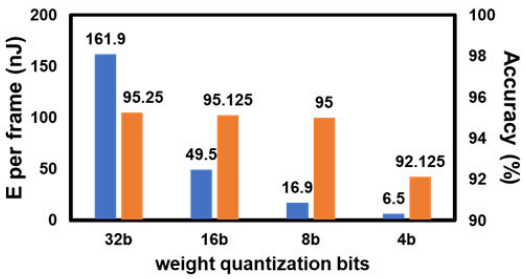


Fig. 4. Simulated energy per frame and accuracy with different quantization bits.

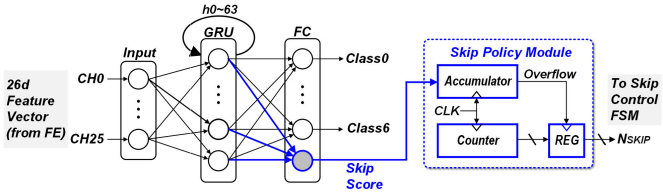


Fig. 5. Structure of the skip-RNN model.

Since the cost of floating-point operations is expensive both in terms of memory size and computation energy [24], we trained our model with different fixed-point bit-widths and tested the corresponding accuracy. We conducted quantization-aware training using a custom fixed-point Python model of our KWS system to minimize the accuracy drop caused by quantization error, ultimately selecting an 8-bit fixed-point representation for the weights (Fig. 4). Furthermore, we used this model to establish our hyperparameters related to the KWS system, such as the sampling rate and dimension of the MFCC feature vector.

To achieve a balance between accuracy and computation energy consumption, we employed the following loss function to train the skip-RNN model:

$$L_{\text{loss}} = L_{\text{acc}} + \lambda \cdot \sum_{t=1}^{T} u_t. \tag{10}$$

Here, the first term represents the cost associated with the accuracy of the model and the second term represents the skipping rate, where the larger values of λ penalize the skip decision. Thus, by tuning the value of λ, the skip ratio can be changed.

The proposed skip-RNN model (Fig. 5) takes the 26-D feature vector per frame (16-ms time step) from the FE and feeds them to the 64 hidden nodes of the RNN, each using a GRU. The 64 hidden states of the RNN are processed by the fully-connected (FC) layer with eight output nodes. Seven of these FC nodes generate KWS class outputs and the last FC node produces the skip score ($\Delta\tilde{u}_t$) for the skip policy module. The skip score is accumulated until the accumulator overflows exceeding 0.5. The register latches the counter output when the overflow occurs to produce N_{SKIP}, which indicates how many frames to skip.

IV. Proposed KWS System Optimization: Hardware Level

A. High-Level System Overview

The operating principle of the proposed KWS system is illustrated in Fig. 6. The AFE generates a 16-ms audio frame

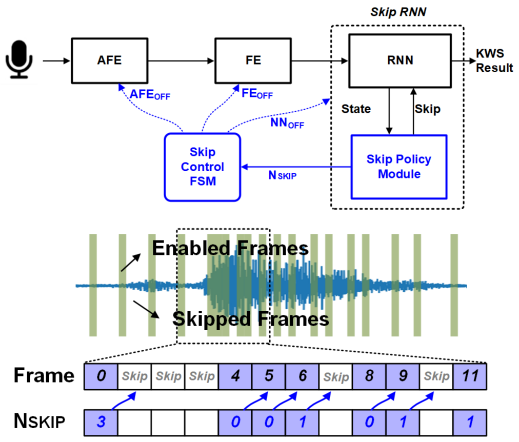


Fig. 6. Proposed KWS system and conceptual operation.

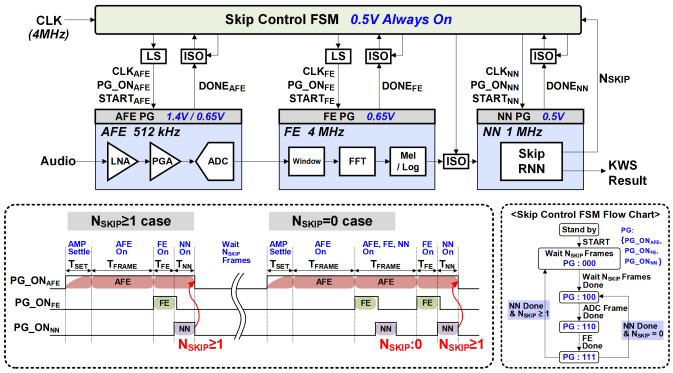


Fig. 7. Detailed block diagram of the proposed KWS SoC.

from the input audio waveform. The frame is then converted into a log-Mel feature vector by the FE and fed to the NN classifier. The skip policy module adaptively determines whether the hidden state is updated or skipped and also generates an N_{SKIP}. As mentioned in the previous section, the skip-RNN, together with the skip policy module, is trained end-to-end by observing a sequence of frames to simultaneously learn both the keyword classification and the skip control. Once a decision is made by the skip policy module, we power down the system, including AFE and FE, for N_{SKIP} frames. The skip control finite-state machine (FSM), which is always on, controls the system power gating.

Fig. 7 shows the overall block diagram of the proposed KWS system, comprised of an AFE, FE, NN classifier, and skip control FSM. The skip control FSM provides PG signals for each of the other three blocks, as well as their corresponding clocks, start, and isolation control signals. KWS system operation starts by turning on the AFE PG with the $\text{PG\_ON}_{\text{AFE}}$ signal, followed by the $\text{START}_{\text{AFE}}$ signal to enable the AFE control logic. The FSM inserts a programmable timing margin (up to 250 μs with 15.6-μs step) between $\text{PG\_ON}_{\text{AFE}}$ and $\text{START}_{\text{AFE}}$ for proper supply stabilization. After the amplifier settling time (T_{SET}), an ADC begins sampling the incoming signal. Upon conversion of one audio frame, the FE and the RNN module process the data. The AFE, therefore, remains on for an additional $T_{\text{FE}} + T_{\text{NN}}$ to avoid losing any data from the next frame while the RNN processes the current frame. When re-enabling the AFE after one or more skipped frames,

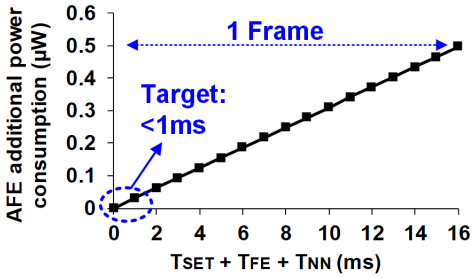


Fig. 8. AFE additional power according to the switching overhead.

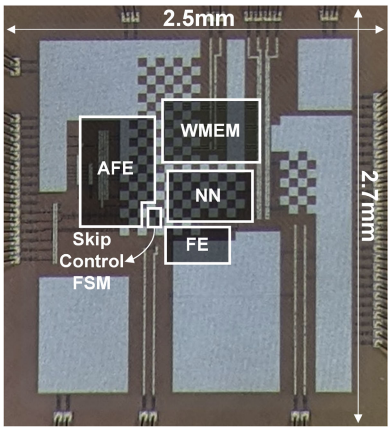


Fig. 9. Photograph of the fabricated chip.

the FSM prestarts the AFE for T_{SET} to stabilize it before enabling ADC to digitize the new frame. The bottom right side of Fig. 7 shows the flowchart of the skip control FSM. In addition to the active time duration T_{FRAME}, the AFE also consumes power during T_{SET}, T_{FE}, and T_{NN}, which represents an energy overhead (Fig. 8). Therefore, to minimize this AFE energy overhead, the proposed design focuses on achieving a fast settling time for the AFE and short operation times for the FE and NN classifiers.

The proposed chip is fabricated in 28-nm CMOS technology with a die photograph in Fig. 9. The area of the chip is 2.5 × 2.7 mm, with a core size of 0.8 mm^2 including an 18-KB ON-chip memory.

B. Analog Frontend

The proposed fast-settling AFE is shown in Fig. 10. A CF amplifier is attractive for its low power consumption but has a high-pass corner (cut-off) frequency (f_{HP}) that is inherently low, resulting in a long settling time (e.g., 37 ms with f_{HP} = 50 Hz in the baseline implementation), which is comparable to T_{FRAME}. We, therefore, designed the first-stage LNA with a dc-coupled Gm-ratio structure [25], while the second-stage programmable gain amplifier (PGA) uses a CF structure. Note that the PGA core shares the same structure as the LNA core but with the exclusion of the diode-connected load (1x branch) shown in Fig. 10. The driver (DRV) core consists of two 5T opamps used as a unit gain feedback structure. The frequency response of each stage is shown in Fig. 11.

Using a dc-coupled structure for the first stage reduces settling time by 54% due to its lack of a high-pass corner.

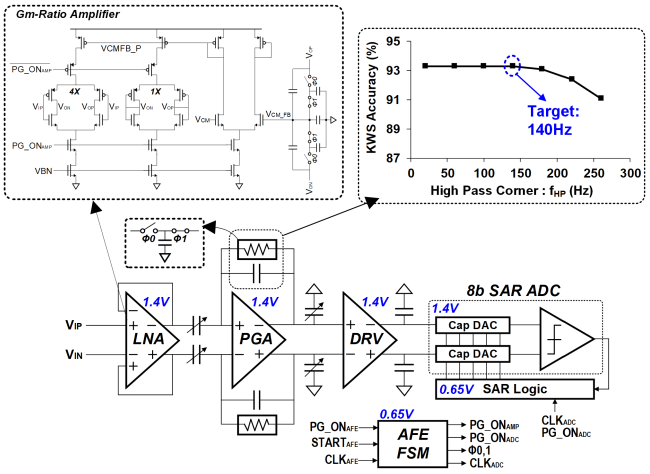


Fig. 10. Proposed fast settling AFE structure.

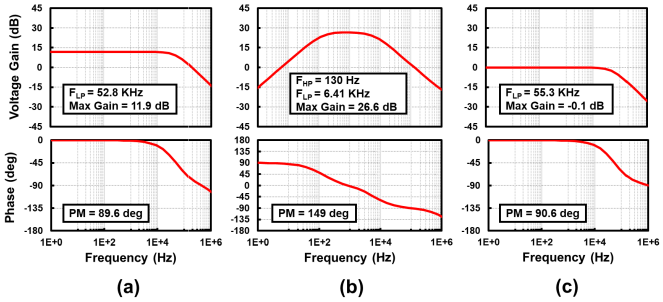


Fig. 11. Simulated frequency response of (a) LNA, (b) PGA, and (c) DRV.

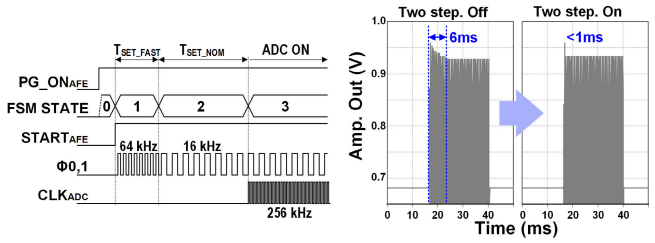


Fig. 12. Proposed two-step frequency control scheme.

To improve the settling time of the CF PGA, its f_{HP} must be set as high as possible without filtering out voice content that would degrade KWS accuracy. Based on software KWS simulations (Fig. 10), f_{HP} can be increased to 140 Hz (compared to a conventional value of <50 Hz) but must be carefully controlled across PVT conditions to avoid inadvertent excursions into higher frequencies that would impact KWS accuracy.

Traditionally, the corner/cut-off frequency in a CF amplifier is set using a pseudo-resistor, but its high PVT sensitivity (>10×) cannot achieve the required corner frequency accuracy. Instead, we employ a switched capacitor resistor with equivalent resistance set by its capacitance and the switching clock ($\Phi_{0,1}$), which are insensitive to PVT. Furthermore, we also adopt a two-step frequency control, where $\Phi_{0,1}$ operates momentarily at a higher frequency to achieve fast settling before switching to its final value (Fig. 12), reducing T_{SET} by 6×. By combining all techniques, the settling time is reduced

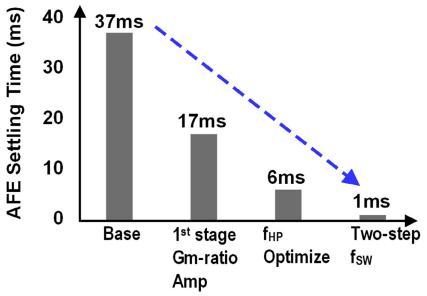


Fig. 13. AFE settling time improvement by each optimization.

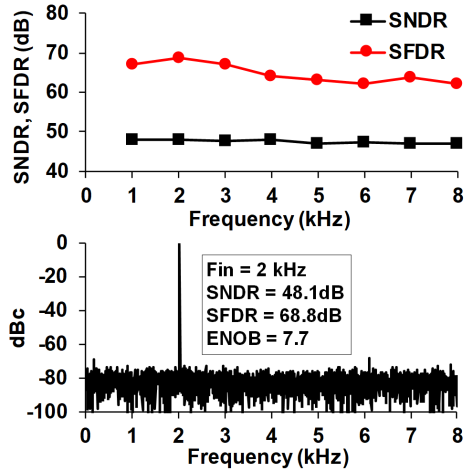


Fig. 14. Measured AFE performance.

from 37 ms to less than 1 ms, yielding a 1.1-μW average power reduction in the final realized system. The settling time improvement by each optimization step is summarized in Fig. 13.

The ADC is an 8-bit synchronous SAR architecture operating at a sampling frequency of 16 kS/s. The amplifiers, the capacitor digital-to-analog converter (DAC), and the ADC comparator operate at a voltage of 1.4 V, while the AFE FSM and the SAR logic operate at 0.65 V. When $\text{PG\_ON}_{\text{AFE}}$ goes low, all AFE blocks, except for the capacitor DAC, are power gated. The measured performance of the AFE is detailed in Fig. 14.

C. Feature Extractor

The FE, shown in Fig. 15, consists of windowing, FFT, Mel filter, and log units. A 16-ms 256-point 8-bit nonoverlapping audio frame is multiplied by the Hanning window function and forwarded to the FFT module. The output is then fed to power calculation, Mel filter, and log modules, generating an 8-bit 26-D feature vector.

To achieve low-power operation, minimizing the power consumption of the FFT module is crucial, as it contributes significantly to FE power. However, since the FFT module must process nonconsecutive frames in the proposed KWS system, a conventional pipelined architecture, which benefits from high throughput and low power, is unsuitable. Therefore, we employ a memory-based fully-folded FFT structure with a single butterfly unit (BFLY; Fig. 15, bottom left).

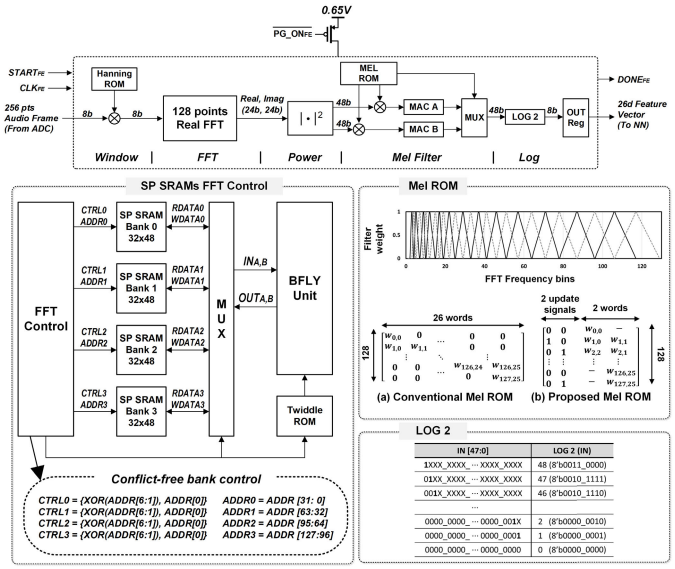


Fig. 15. Structure of the feature extraction block.

Instead of a conventional single-bank dual-port SRAM for concurrent read and write operations, we use four banks of a single-port SRAM which has lower dynamic power consumption owing to the simpler/smaller memory periphery circuitry. Along with a conflict-free bank scheduling (Fig. 15, bottom), we can achieve a 70% reduction of the memory power consumption without reducing the data access rate. Additionally, we replaced a 256-point complex FFT on the real-valued input with a 128-point FFT for the mathematically equivalent output [26], reducing the number of twiddle factor multiplication from 1024 to 576, and the SRAM size by half.

In addition to the optimization of the FFT module, we also focused on reducing the hardware cost in the filter bank implementations (Fig. 15 bottom right). First, we replaced the floating-point natural logarithm operation with a hardware-friendly base-2 logarithm operation and adopted a leading-one detector-based lookup table (LUT) with a 48-bit input and 8-bit output instead of the conventional coordinate rotation digital computer (CORDIC)-based log operation [27]. For instance, if the input to the log unit is 48'b0000...000111 (d'7) as shown in Fig. 15 (LOG2), the accurate log2 value is 2.807, whereas our implementation produces the output of 3 (8'b00000011) which is a ceiling approximation of the actual value. This results in hardware cost savings without a significant impact on the KWS accuracy.

Moreover, we reduced the memory size of the Mel filter operation by employing a special LUT that contains two columns for the update signals and two columns for storing Mel filter weights for the even and odd filter bands. Two multiply-and-accumulate (MAC) units, MAC A and B in Fig. 15, calculate the even and odd Mel filter bands with the stored weight vectors, and when there is an update signal, the MAC units output their value and reset. While a naive direct LUT implementation needs a weight matrix of 128 $\times$ 26 words storing 8-bit weights per word [Fig. 15 MEL ROM (a)], our method reduces the LUT size to 128 $\times$ 4 words [Fig. 15 MEL ROM (b)] where each row consists of two 1-bit MAC update

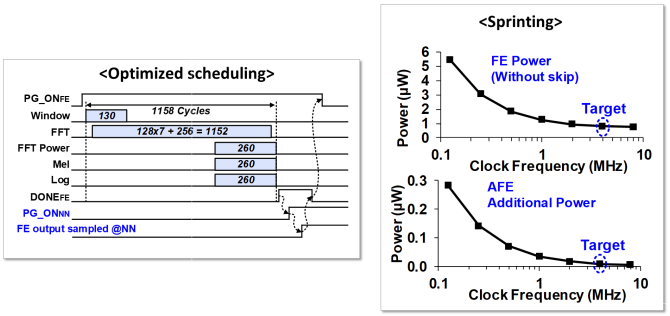


Fig. 16. Operation time optimization of the FE.

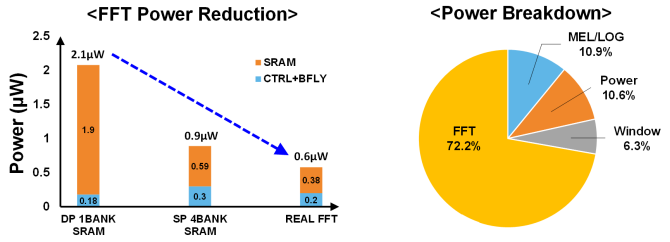


Fig. 17. Power reduction by FFT optimization and power breakdown of the FE.

signals and two 8-bit weights. This optimization achieves a 12× reduction in the Mel filter weight storage.

The left side of Fig. 16 shows the optimized timing diagram of the FE. One BFLY calculation requires two cycles, where one additional cycle is used to register the BFLY input data to reduce glitch power, saving 22% BFLY power. The architecture requires 1152 cycles to complete one FFT and perform the window overlapping (apodization), power, Mel, and log operations. To reduce overall operating time, the FE *sprints* with a high clock frequency (4 MHz) to complete its computation quickly (290 μs, much shorter than the frame length of 16 ms) to reduce AFE power overhead when skip decision is made. Upon completion, the FE is then power gated to minimize its leakage power for the remaining time within a frame. The FE sprinting frequency is set at 4 MHz, which marks the point of diminishing returns as seen in the right side of Fig. 16. The power reduction with each FFT optimization (four-bank single-port SRAMs, real-valued FFT) and overall power breakdown of the FE are shown in Fig. 17.

D. NN Accelerator

Fig. 18 shows the detailed implementation of the skip-RNN accelerator, which consists of a MAC array, WMEM, functional units (FUs), register banks, and an NN FSM. The MAC array is designed with 64 parallel MACs. We adopt an output stationary dataflow for the MAC array, where the partial sums for the 64 GRU gate vectors are accumulated in the 64 MACs until the processing of each GRU gate vector is completed. For the FC layer, eight MACs process the eight FC output nodes in parallel. All activations are scaled and quantized to 12 bits. Nonlinear functions in FU, such as hyperbolic tangent and sigmoid, are implemented using a single LUT with rescaling, which is more energy-efficient than on-the-fly calculations. The always-on WMEM stores the 8-bit weights

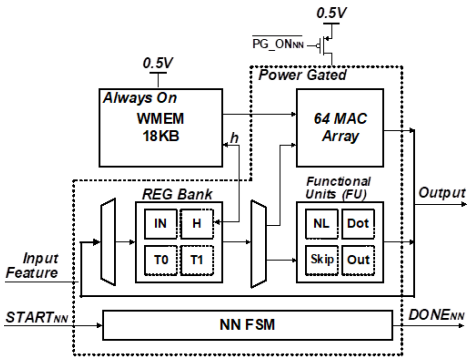


Fig. 18. Detailed structure of the proposed skip-RNN accelerator.

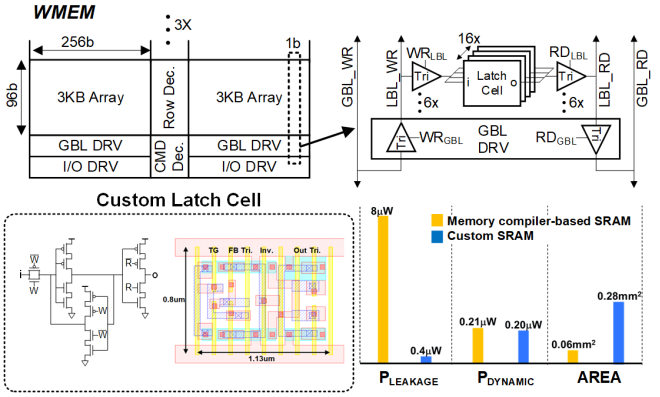


Fig. 19. Structure of low-leakage custom WMEM and performance comparison with the memory compiler-based SRAM.

Time

NN #Phase / Operation		0 : r	1 : z	2 : $\tilde{h}$	3 : Out	4 : FC	5 : Skip
#Cycle		95	92	92	65	84	67
Processing Unit (Dest. Reg.)	MAC	T0	T0	T0		T0	
	NL		T1	T1	Out		
	Dot			MAC			
	Out				H		
	Skip						Done
Register (Dest. P.U.)	IN	MAC	MAC	MAC			
	H	MAC	MAC	Dot	Out	MAC	
	T0		NL	NL	NL		Skip
	T1			Dot	Out		

Fig. 20. Scheduling table of the RNN accelerator.

for the RNN model and is custom-designed using latches with high-Vth transistors. Although there is a 4.7× increase in area, the custom SRAM achieves 20× leakage power reduction compared to the memory compiler-based SRAM (Fig. 19) [28]. Without this leakage power reduction using the custom latch-based memory, the overall system power consumption would be dominated by the leakage from the always-on RNN WMEM (8 μW).

After each frame, the RNN hidden states are stored in WMEM before the NN block is power-gated. The operation of the MAC array, FU, and register bank accesses are scheduled to overlap their operations reducing a single frame processing time to 495 cycles (Fig. 20). The RNN sprints with a 1-MHz clock to reduce the leakage power and optimize overall energy efficiency by early power gating for the remaining time within a frame.

V. Measurement Results and Discussions

A. Test Procedures

We employed the Google speech command dataset (GSCD) v1 [29] to train and test the proposed KWS system. The

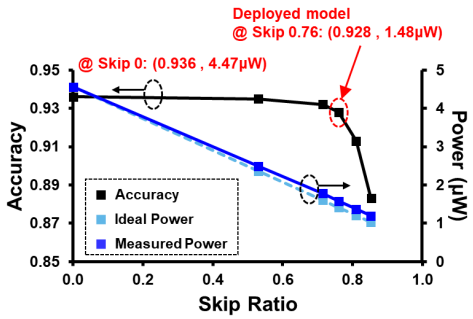


Fig. 21. Accuracy and power reduction tradeoff by different skip ratios (seven class KWS).

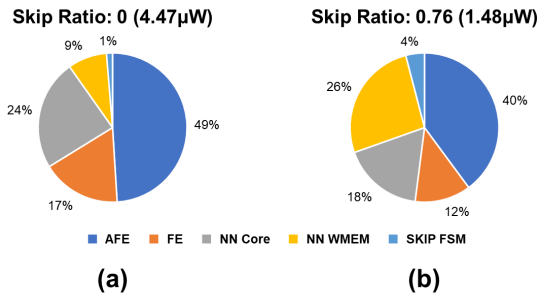


Fig. 22. Power breakdown for (a) without skipping and (b) deployed model.

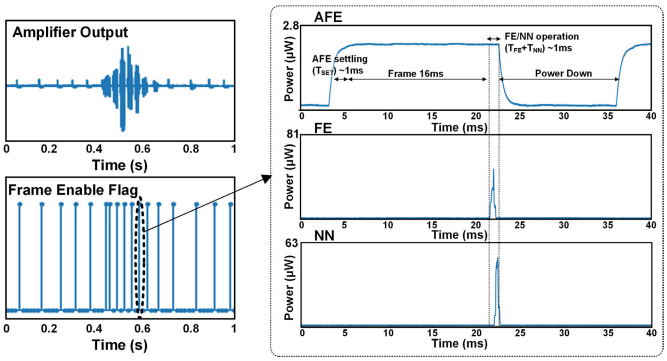


Fig. 23. Measured waveform of the proposed KWS system.

GSCD consists of 65 000 one-second length audio clips with 12 classes, including ten one-word commands, out-of-vocabulary (oov), and noise. To evaluate our system, we used five keywords ("yes," "down," "marvin," "right," and "zero") from the GSCD along with oov and noise. We used 1000 utterances per keyword for training and 250 utterances per keyword for testing. The network was trained with a Python model of our KWS system using TensorFlow. To measure the proposed KWS system on chip (SoC), we used TI 8550 DACs to generate analog audio input signals to the chip and a general purpose input output (GPIO) board to communicate and control the chip, along with other external digital and analog supplies.

B. KWS Performance Evaluation

We evaluated the performance of the proposed KWS SoC with seven classes (five keywords, oov, and noise) by training the model with different skip ratios (Fig. 21). As expected, as the skip ratio increases, the accuracy decreases and the overall system power decreases, mapping a tradeoff space. Until the skip ratio of 0.76, the accuracy drop was around or less than 1%; however, beyond that point, there is significant accuracy degradation. Since we minimized the switching overhead of the system and the power consumption of always-on blocks, the actual power saving is close to that of the ideal case that linearly scales with the skip ratio as shown in Fig. 21.

With an optimal skip ratio of 0.76, we achieve an accuracy of 0.928 which is less than a 1% drop from the baseline while achieving 3× power reduction. The measured power consumption of each block is depicted in Fig. 22, where the overall power consumption of the system is reduced from 4.47 [Fig. 22(a)] to 1.48 μW [Fig. 22(b)]. Additionally, Fig. 23 displays the measured amplifier output and frame enable control for an example audio signal, demonstrating the adaptive enable/skip pattern which is input dependent. Fig. 23 also shows the measured dynamic power of the AFE, FE, and NN during operation.

C. Comparison With State-of-the-Art Works

Table I provides a comparison with recently published KWS chips. The proposed work achieves 92.8% KWS accuracy with seven classes using the GSCD dataset. The AFE, FE, and NN power consumption are 0.59, 0.18, and 0.71 μW, respectively, achieving a 1.48-μW total power consumption. The proposed design is the lowest power KWS system that fully integrates AFE, FE, and NN.

In terms of accuracy, our system shows the second-highest among the compared works. The only work that has better accuracy [7], uses three classes for evaluation with a small-sized (2 KB) NN WMEM, which explains their low digital power consumption. However, their accuracy drops significantly for tasks with more than five classes (<92%) which is lower than our KWS accuracy with seven classes. We also tested four class KWS tasks for a fair comparison with [7], [16], and [30], where we achieved 93.6% KWS accuracy and 1.37-μW total power consumption with a skip ratio of 0.8. Although we have not implemented the 12 classes case in hardware to save associated WMEM cost, our simulations have shown that the accuracy degradation compared to the seven classes case is around 3% with the same skip ratio (0.76) and expected power consumption is slightly over 1.5 μW. Considering this, the expected tested accuracy would be similar to that of [9], but with much lower power consumption than both [8] and [9].

D. Discussions

Based on the measured results, we have identified further optimization opportunities for future work. For instance, considering the relatively low sampling rate of the audio signal that we use (16 kS/s), the number of MFCC filter coefficients can be reduced. Simulation results in Fig. 24 demonstrate that the 20 MFCC filter coefficients case outperforms that of 26 with the same sampling rate because using fewer coefficients is more robust to minor frequency/pitch variations of different speakers. This explains the high accuracy of [7] which only uses an 8 kS/s sampling rate and ten MFCC filter

TABLE I
STATE-OF-THE-ART COMPARISON

		This work	ISSCC 2022 [8]	ISSCC 2021 [30]	ISSCC 2020 [7]	VLSI 2019 [9]	ESSCIRC 2018 [16]
Technology [nm]		**28**	65	65	28	65	65
Area [mm^2]		**0.8**	2.03	0.72	0.23	2.56	1.04
Normalized Area[1]		**1**	0.47	0.17	0.29	0.59	0.24
Algorithm		**skip-RNN**	RNN	SNN	DSCNN	RNN	RNN
Memory [KB]		**18**	27	-	2	105	32
Latency [ms]		**16**	12.4	100	64	16	16
Analog Frontend		**YES**	YES	NO	NO	YES	NO
Feature Extraction		**YES**	YES	YES	YES	YES	NO
Neural Network		**YES**	YES	YES	NO	YES	YES
Dataset		**GSCD**	GSCD	GSCD	GSCD	GSCD	N/A
Number of Classes (Keywords)		**7 (5)**	12 (10)	4 (4)	3 (2)	12 (10)	4 (4)
Accuracy [%]		**92.8**	86.0	90.2	94.6	90.9	91.2
Power [µW]	AFE	**0.59**	13	-	-	1.8	-
	FE	**0.18**		0.11	0.34	7.1	-
	NN	**0.71**	10	0.1	0.17	3.4	5
	Total	**1.48**	23	-	-	16.1	-

[1] Normalized Area/F^2

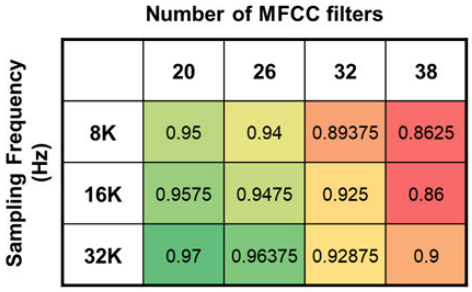


Fig. 24. Colormap of accuracies in terms of different sampling frequencies and the number of MFCC filters based on simulation result with four class KWS.

coefficients in their FE (in addition to their benefits from having a small number of keywords and utilizing the DS-CNN algorithm which is known to have better than RNNs as mentioned in the Section II). Furthermore, we have observed from simulations that the accuracy of the 8 kS/s is similar to that of the 16 kS/s when using an equal number of MFCC filter coefficients. This means that we can also reduce the AFE power consumption as well by designing it for a lower sampling rate. Also, utilizing an analog FE as in [30] can be another promising option to further optimize the power consumption.

VI. CONCLUSION

A low-power fully integrated KWS SoC with content-adaptive frame subsampling fabricated in 28-nm CMOS technology is proposed in this work. The overall system is co-optimized from end-to-end, including AFE, digital back-end, and algorithms. To fully exploit the benefit of the adaptive frame subsampling technique, the following design considerations are made. First, a fast-settling AFE structure featuring a dc-coupled first stage with the switched CF resistor is utilized to enable rapid on/off. Additionally, the FE and RNN accelerators are designed to complete their operations with optimized operation cycles and sprint with a relatively fast clock. Lastly, each block is customized for low power. In particular, various techniques for the leakage power reduction of the always-on block are employed to maximize power savings from the dynamic power gating. As a result, the proposed work demonstrates 1.48 μW with an average of 76% skip ratio across frames, achieving 92.8% accuracy using the GSCD dataset.

ACKNOWLEDGMENT

We gratefully acknowledge the TSMC University Shuttle Program for chip fabrication. This work was supported in part by COGNISENSE, one of seven centers in JUMP 2.0, a Semiconductor Research Corporation (SRC) program sponsored by DARPA.

REFERENCES

[1] J.-H. SeoI et al., "A 1.5μW end-to-end keyword spotting SoC with content-adaptive frame sub-sampling and fast-settling analog frontend," in *Proc. IEEE Int. Solid- State Circuits Conf. (ISSCC)*, Feb. 2023, pp. 1–3.

[2] I. López-Espejo, Z.-H. Tan, J. H. L. Hansen, and J. Jensen, "Deep spoken keyword spotting: An overview," *IEEE Access*, vol. 10, pp. 4169–4199, 2022.

[3] J. S. P. Giraldo and M. Verhelst, "Hardware acceleration for embedded keyword spotting: Tutorial and survey," *ACM Trans. Embedded Comput. Syst.*, vol. 20, no. 6, pp. 1–25, Oct. 2021.

[4] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[5] S. Bang et al., "14.7 A 288μW programmable deep-learning processor with 270KB on-chip weight storage using non-uniform memory hierarchy for mobile intelligence," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, pp. 250–251, Feb. 2017.

[6] M. Price, J. Glass, and A. P. Chandrakasan, "A low-power speech recognizer and voice activity detector using deep neural networks," *IEEE J. Solid-State Circuits*, vol. 53, no. 1, pp. 66–75, Jan. 2018.

[7] W. Shan et al., "14.1 A 510nW 0.41 V low-memory low-computation keyword-spotting chip using serial FFT-based MFCC and binarized depthwise separable convolutional neural network in 28nm CMOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2020, pp. 230–232.

[8] K. Kim et al., "A 23μW solar-powered keyword-spotting ASIC with ring-oscillator-based time-domain feature extraction," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2022, pp. 1–3.

[9] J. S. P. Giraldo, S. Lauwereins, K. Badami, H. Van Hamme, and M. Verhelst, "18μW SoC for near-microphone keyword spotting and speaker verification," in *Proc. Symp. VLSI Circuits*, Jun. 2019, pp. C52–C53.

[10] L. Ye et al., "The challenges and emerging technologies for low-power artificial intelligence IoT systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 12, pp. 4821–4834, Dec. 2021.
[11] V. Campos, B. Jou, X. G. I. Nieto, J. Torres, and S.-F. Chang, "Skip RNN: Learning to skip state updates in recurrent neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2018.
[12] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.
[13] P. K. Chundi et al., "Always-on sub-microwatt spiking neural network based on spike-driven clock- and power-gating for an ultra-low-power intelligent device," *Frontiers Neurosci.*, vol. 15, Jul. 2021, Art. no. 684113.
[14] Z. Wang et al., "12.1 A 148nW general-purpose event-driven intelligent wake-up chip for AIoT devices using asynchronous spike-based feature extractor and convolutional neural network," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2021, pp. 436–438.
[15] W. Shan, J. Qian, L. Zhu, J. Yang, C. Huang, and H. Cai, "AAD-KWS: A sub-μw keyword spotting chip with an acoustic activity detector embedded in MFCC and a tunable detection window in 28-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 58, no. 3, pp. 867–876, Mar. 2023.
[16] J. S. P. Giraldo and M. Verhelst, "Laika: A 5μW programmable LSTM accelerator for always-on keyword spotting in 65nm CMOS," in *Proc. IEEE 44th Eur. Solid State Circuits Conf. (ESSCIRC)*, Sep. 2018, pp. 166–169.
[17] Q. Li et al., "NS-KWS: Joint optimization of near-sensor processing architecture and low-precision GRU for always-on keyword spotting," in *Proc. ACM/IEEE Int. Symp. Low Power Electron. Design*, Aug. 2020, pp. 97–102.
[18] Q. Li et al., "NS-FDN: Near-sensor processing architecture of feature-configurable distributed network for beyond-real-time always-on keyword spotting," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 5, pp. 1892–1905, May 2021.
[19] Y. Zhang, N. Suda, L. Lai, and V. Chandra, "Hello edge: Keyword spotting on microcontrollers," 2018, *arXiv:1711.07128v3*.
[20] *T3902 Bottom Port PDM Low-Power Multi-Mode Microphone With High AOP Mode*, TDK InvenSense, 2020, rev. 1.0. https://invensense.tdk.com/wp-content/uploads/2020/05/DS-000357-T3902-v1.0.pdf
[21] *T5838 Bottom Port PDM Digital Output Multi-Mode Microphone*, Rev. 1.0, TDK InvenSense, San Jose, CA, USA, 2022. [Online]. Available: https://invensense.tdk.com/wp-content/uploads/2022/08/DS-000383-T5838-v1.0.pdf
[22] S. Oh, T. Jang, K. D. Choo, D. Blaauw, and D. Sylvester, "A 4.7μW switched-bias MEMS microphone preamplifier for ultra-low-power voice interfaces," in *Proc. Symp. VLSI Circuits*, Jun. 2017, pp. C314–C315.
[23] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*.
[24] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized neural networks: Training neural networks with low precision weights and activations," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 6869–6898, 2017.
[25] J. Lim et al., "A 0.19×0.17mm^2 wireless neural recording ic for motor prediction with near-infrared-based power and data telemetry," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Sep. 2020, pp. 416–418.
[26] T. Holton, *Digital Signal Processing: Principles and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2021.
[27] W. Shan et al., "A 510-nW wake-up keyword-spotting chip using serial-FFT-based MFCC and binarized depthwise separable CNN in 28-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 56, no. 1, pp. 151–164, Jan. 2021.
[28] S. Hanson et al., "A low-voltage processor for sensing applications with picowatt standby mode," *IEEE J. Solid-State Circuits*, vol. 44, no. 4, pp. 1145–1155, Apr. 2009.
[29] P. Warden. *Speech Commands: A Public Dataset for Single-Word Speech Recognition*. Dataset. Accessed: May 5, 2021. [Online]. Available: http://download.tensorflow.org/data/speech_commands_v0.01.tar.gz
[30] D. Wang, S. J. Kim, M. Yang, A. A. Lazar, and M. Seok, "9.9 A background-noise and process-variation-tolerant 109nW acoustic feature extractor based on spike-domain divisive-energy normalization for an always-on keyword spotting device," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 64, Feb. 2021, pp. 160–162.

Heejin Yang (Graduate Student Member, IEEE) received the B.S. and M.S. degrees in electrical and computer engineering from Seoul National University, Seoul, South Korea, in 2019 and 2021, respectively. She is currently pursuing the Ph.D. degree in electrical and computer engineering with the University of Michigan, Ann Arbor, MI, USA.

Her current research interests include ultralow-power VLSI design and computing systems.

Ji-Hwan Seol (Member, IEEE) received the B.S. degree in electrical engineering from Yonsei University, Seoul, South Korea, in 2009, and the M.S. degree in electrical engineering from KAIST, Daejeon, South Korea, in 2012, and the Ph.D. degree in electrical and computer engineering from the University of Michigan, Ann Arbor, MI, USA, in 2022. He has been with Samsung Electronics since 2012. His research interests include memory systems, frequency synthesizers, and deep-learning hardware.

Rohit Rothe (Graduate Student Member, IEEE) received the dual B.Tech. and M.Tech. degrees in electrical engineering from the IIT Bombay, Mumbai, India, in 2018. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the University of Michigan, Ann Arbor, MI, USA.

His current research interests include ultralow-power analog VLSI design, low power dc–dc converters, and Internet-of-Things (IoT) sensor systems.

Zichen Fan (Graduate Student Member, IEEE) received the B.S. degree from Tsinghua University, Beijing, China, in 2019. He is currently pursuing the Ph.D. degree with the Michigan Integrated Circuit Laboratory, University of Michigan, Ann Arbor, MI, USA.

His current research interests include machine-learning accelerator design, algorithm-hardware codesign, and energy-efficient digital system design.

Qirui Zhang (Member, IEEE) received the B.S. degree (Hons.) from the School of Microelectronics, Shanghai Jiao Tong University, Shanghai, China, in 2018. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the University of Michigan, Ann Arbor, MI, USA.

His research interests are in efficient algorithm-hardware codesigns, VLSI architectures, and integrated circuits for emerging applications, including artificial intelligence and quantum computing.

Mr. Zhang was a recipient of the Best Paper Award at the 2022 tinyML Research Symposium, and the 2023 IEEE International Conference on Application-specific Systems, Architectures, and Processors. He was a Finalist of the 2023 Qualcomm Innovation Fellowship (North America).

Hun-Seok Kim (Senior Member, IEEE) received the B.S. degree in electrical engineering from Seoul National University, Seoul, South Korea, in 2001, and the Ph.D. degree in electrical engineering from the University of California at Los Angeles, Los Angeles, CA, USA, in 2010.

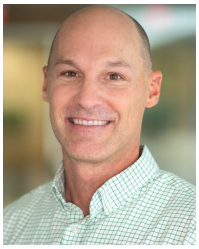
He is currently an Associate Professor with the University of Michigan at Ann Arbor, Ann Arbor, MI, USA. His research focuses on system analysis, novel algorithms, and VLSI architectures for low-power/high-performance wireless communications, signal processing, computer vision, and machine-learning systems.

Dr. Kim is a recipient of the DARPA Young Faculty Award in 2018 and the National Science Foundation (NSF) CAREER Award in 2019. He is an Associate Editor of IEEE TRANSACTIONS ON MOBILE COMPUTING.

David Blaauw (Fellow, IEEE) received the B.S. degree in physics and computer science from Duke University, Durham, NC, USA, in 1986, and the Ph.D. degree in computer science from the University of Illinois at Urbana-Champaign, Urbana, IL, USA, in 1991.

Until August 2001, he worked for Motorola, Inc. with Austin, TX, USA, where he was the Manager of the High Performance Design Technology Group and won the Motorola Innovation Award. Since August 2001, he has been on the faculty of the University of Michigan, Ann Arbor, MI, USA, where he is the Kensall D. Wise Collegiate Professor of Electrical Engineering and Computer Science (EECS). He has published over 600 articles, has received Numerous Best Paper Awards, and holds 65 patents. He has researched ultralow-power wireless sensors using subthreshold operation and low-power analog circuit techniques for millimeter systems. This research was awarded the MIT Technology Review's "one of the year's most significant innovations." His research group introduced so-called near-threshold computing, which has become a common concept in semiconductor design. Most recently, he has pursued research in cognitive computing using analog, in-memory neural networks for edge devices and genomics for precision health.

Dr. Blaauw was the General Chair of the IEEE International Symposium on Low Power and a member of the IEEE International Solid-State Circuits Conference (ISSCC) analog program subcommittee. He is an IEEE Fellow and received the 2016 SIA-SRC Faculty Award for lifetime research contributions to the U.S. semiconductor industry.



Dennis Sylvester (Fellow, IEEE) received the Ph.D. degree in electrical engineering from the University of California, Berkeley, CA, USA, where his dissertation was recognized with the David J. Sakrison Memorial Prize as the most outstanding research in the UC-Berkeley Electrical Engineering and Computer Science (EECS) department.

He is the Edward S. Davidson Collegiate Professor and Interim Chair of Electrical and Computer Engineering with the University of Michigan, Ann Arbor, MI, USA, and was the founding Director of the Michigan Integrated Circuits Laboratory (MICL), a group of ten faculty and 70+ graduate students. He has held research staff positions in the Advanced Technology Group of Synopsys, Mountain View, CA, USA, Hewlett-Packard Laboratories with Palo Alto, CA, USA, and Visiting Professorships with the National University of Singapore, Singapore, and Nanyang Technological University, Singapore. He has published over 550 articles along with one book and several book chapters. He holds 51 U.S. patents. He co-founded Ambiq, a fabless semiconductor company developing ultralow power mixed-signal solutions for compact wireless devices. His research interests include the design of millimeter-scale computing systems and energy-efficient near-threshold computing.

Dr. Sylvester received the National Science Foundation (NSF) CAREER Award, the Beatrice Winner Award at International Solid-State Circuits Conference (ISSCC), an IBM Faculty Award, an SRC Inventor Recognition Award, and 16 Best Paper Awards and nominations. He was named a top five Contributing Author all-time at ISSCC in 2023, the most prolific author at the IEEE Symposium on very large-scale integration (VLSI) Circuits, and was awarded the University of Michigan Henry Russel Award for distinguished scholarship. He is the current Editor-in-Chief of the IEEE JOURNAL OF SOLID-STATE CIRCUITS. He previously served on the administrative committee for the IEEE Solid-State Circuits Society, was an Associate Editor for IEEE JOURNAL OF SOLID-STATE CIRCUITS, IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN, and IEEE TRANSACTIONS ON VERY-LARGE-SCALE INTEGRATION SYSTEMS, and served as an IEEE Solid-State Circuits Society Distinguished Lecturer. He is a Fellow of the National Academy of Inventors.